On the Similarity between Attention and SVM on the Token Separation and Selection Behavior

Anonymous authors
Paper under double-blind review

Abstract

The attention mechanism underpinning the transformer architecture is effective in learning the token interaction within a sequence via softmax similarity. However, the current theoretical understanding on optimization dynamics of the softmax attention is insufficient in characterizing how attention performs intrinsic token separation and selection, which is crucial to sequence-level understanding tasks. On the other hand, support vector machines have been well-studied of its max-margin separation behaviour. In this paper, we will formulate the softmax attention convergence dynamics as hard-margin SVM optimization problem. We adopt a tensor trick to formulate the matrix-based attention optimization problem and relax the strong assumptions on the derivative of the loss function from the prior works. As a result, we demonstrate that gradient descent converges to the optimal solution for SVM. In addition, we show softmax is more stable than other linear attention through analysis on their lipschitz. Our theoretical insights are validated through numerical experiments, shedding insights on the convergence dynamics of softmax attention as the foundational stones on the success of the large language models.

1 Introduction

The transformer architecture was introduced and traditionally consists of alternating attention and multilayer-perceptron (MLP) sublayers, has given rise to influential models in the realm of complex natural language tasks. These models include BERT Devlin et al. (2018), RoBERTa Liu et al. (2019), XLNet Yang et al. (2019), GPT-3 Brown et al. (2020), OPT Zhang et al. (2022a), Llama Touvron et al. (2023), and PaLM Chowdhery et al. (2022). Among these, GPT series, with hundreds of billion parameters, have served as fundamental building block to power ChatGPT OpenAI (2022), a chat software capable of generating highly convincing textual responses, creating immersive user experiences. In addition, the arrival of the next-generation GPT-4 OpenAI (2023) has expanded the horizons of AI capabilities, enabling it to excel in tasks previously considered beyond its reach, achieving human-level proficiency in various professional and academic benchmarks. While widely studied, it is still very challenging to understand the training dynamics of transformer models.

In this paper, we aim to understand the training dynamics of the core component of transformers, known as attention Vaswani et al. (2017); Radford et al. (2018); Devlin et al. (2018); Brown et al. (2020); Alman & Song (2023); Zandieh et al. (2023); Brand et al. (2023); Gao et al. (2023b). It involves projecting tokens into queries, keys, and values and then comparing queries with keys to calculate attention scores. This attention matrix is a dynamic structure that guides the model in assigning importance to individual tokens within a text, allowing it to focus more on tokens relevant to its predictions while downplaying the significance of less informative tokens. This capability has proven invaluable across various domains, including NLP Devlin et al. (2018); Brown et al. (2020); Raffel et al. (2020), computer vision Parmar et al. (2018); Cornia et al. (2020), and reinforcement learning Chen et al. (2021b); Wu et al. (2022). During training, the attention matrix is learned, enabling the model to allocate additional attention to pivotal tokens. To delve into the specifics, the attention matrix computation begins with the multiplication of the query matrix XQ and the key matrix XK, followed by the application of a softmax function to generate a matrix with values ranging between 0 and 1. These values signify the relative importance of each element in the input sequence. A final softmax operation results in the attention matrix. It's obvious that the formulation of the self-attention is a special instance of cross-attention. Given input sequences $A_1, A_2 \in \mathbb{R}^{n \times d}$, we provide the general formulation of the attention computation as the following:

$$\mathrm{Att}(X, Y) = D(X)^{-1} \exp(A_1 X A_2^\top) A_2 Y$$

where $X \in \mathbb{R}^{d \times d}$ denotes the combined parameter $X := QK^\top$, $Q \in \mathbb{R}^{d \times d}, K \in \mathbb{R}^{d \times d}$ are trainable parameter key and query matrices, $Y \in \mathbb{R}^{d \times d}$ denotes the value matrix and $D(X) := \mathrm{diag}(\exp(A_1 X A_2^\top) \mathbf{1}_n) \in \mathbb{R}^{d \times d}$.

We delve into the language modeling task that applied in many popular pretrained models such as BERT Devlin et al. (2018), RoBERTa Liu et al. (2019), GPT-3 Brown et al. (2020). We work with a dataset denoted as $\{A_{l_0,1}, A_{l_0,2}, B_{l_0}\}_{l_0=0}^m$ (for language modeling task, we have $A_{l_0,1} = A_{l_0,2}$), comprising labeled instances B_{l_0}, where each row represents a one-hot vector corresponding in the target sentence. In practice, logistic loss or cross entropy loss is commonly used to do NLP tasks to help classify the next token generation.

Inspired from Tarzanagh et al. (2023a) we give the general form of empirical risk minimization with a logistic loss function as the following:

$$\min_{X \in \mathbb{R}^{d \times d}} L(X) = \min_{X \in \mathbb{R}^{d \times d}} \mathrm{logistic}(\mathrm{Att}(X) \cdot H, B)$$

where $H \in \mathbb{R}^{d \times V}$ denotes the linear prediction head and $B \in \mathbb{R}^{n \times V}$ denotes the labels. As H is a fixed linear prediction head that does not impact our analysis, we will ignore that in the theoretical analysis of our paper.

In addition, for the regression task such as sentiment analysis, we give the formulation of optimization problem of ℓ_2 loss as the following:

$$\min_{X \in \mathbb{R}^{d \times d}} L(X) = \min_{X \in \mathbb{R}^{d \times d}} \| D(X)^{-1} \exp(A_1 X A_2^\top) A_3 Y - B \|_2^2$$

We first convert the matrix representation into a vector representation by employing the well-known tensor trick Diao et al. (2018; 2019); Zhang (2022). This transformation condenses the multiple regression into a single regression task, entailing the rearrangement and grouping of all the elements. Then, we have the simplified optimization problem:

$$\min_{x \in \mathbb{R}^{d^2}} L(x) = \min_{x \in \mathbb{R}^{d^2}} \| \mathrm{mat}(D(x)^{-1} \exp(\mathsf{A}\, x)) A_3 Y - B \|_2^2$$

where $\mathsf{A} := A_1 \otimes A_2$ and $x = \mathrm{vec}(X) \in \mathbb{R}^{d^2}$.

Our main contributions are as follows:

- **SVM equivalence** We characterize the optimization of attention layer by connecting it with a hard max-margin SVM problem (Att-SVM). We show that $W = QK^\top$ trained by gradient descent in the language modeling pretraining converge to the solution of SVM with the Frobenius norm objective.
- **Token selection and contextual sparsity** Building upon the inherent similarity between SVM and attention mechanisms, our experiments show that only a few tokens, which can be regarded as feature vectors, contribute to the gradient computation for each update. Tokens that do not serve as support vectors have zero gradients and can be safely disregarded during pretraining. This property help elucidates the practical effectiveness of sparse training techniques, as evidenced in prior studies Choromanski et al. (2020); Roy et al. (2021); Liu et al. (2023).
- **Token separation** Inspired by the property of SVM, we show that all tokens are gradually separated during the training process. For each subsequent in language modeling task, the decision hyperplane are fundamentally different as the optimal tokens are different for each next work prediction.
- **Lipschitz of Loss Functions** By computing the Lipschitz of different loss functions (logistic loss and ℓ_2-loss) with softmax attention and linear attention, we show that the training of softmax attention is more stable than linear attention.

2 Related works

In this section, we introduce background of Transformer theory as well as SVM, which are critical for our analysis later.

Transformer Theory Previous research has established that the exceptional performance of Transformer-based models can be ascribed to the rich information embedded within their constituent elements, particularly multi-head attention mechanisms. Various studies (Hewitt & Liang, 2019; Tenney et al., 2019; Belinkov, 2022) have presented empirical proof that these components carry a substantial amount of information, making them valuable for tackling a diverse range of probing tasks.

Recent research has explored the potential of Transformer models through a combination of theoretical and experimental approaches. These investigations have delved into several aspects, including their Turing completeness (Bhattamishra et al., 2020b), their capacity for function approximation (Yun et al., 2020; Chen et al., 2021a), their ability to represent formal languages (Bhattamishra et al., 2020a; Ebrahimi et al., 2020; Yao et al., 2021), and their aptitude for learning abstract algebraic operations (Zhang et al., 2022b). Some of these studies the theoretical analysis of attention in different application such as in-context learning Li et al. (2023); Gao et al. (2023b), contextual sparsity prediction Liu et al. (2023).

There are also many methods have been proposed to speedup the computation of Transformer from theoretical perspective. In Brand et al. (2023), they focus on dynamic attention computation and proposed an algorithm that is conditionally optimal, unless the hinted matrix vector multiplication conjecture is proven false. They integrate lazy update methods into their attention computation approach and use the Hinted Matrix-Vector Conjecture to demonstrate the inherent difficulty of this problem. In contrast, other studies Zandieh et al. (2023); Alman & Song (2023) focused on static attention computation. Specifically, Alman & Song (2023) delved into static attention and introduced an algorithm that assessed its complexity within the framework of the exponential time hypothesis.

Panigrahi et al. (2023) introduce innovative techniques that approximate self-attention back propagation that allow a transformer in transformer model to simulate and fine-tune a transformer model within a single forward pass. Malladi et al. (2023) proposed a memory-efficient zero-th order optimizer and theoretically show that adequate pre-training ensures the per-step optimization rate and global convergence rate of their model. Zhao et al. (2023) shows that attention models implicitly approximate parsing to achieve low masked language modeling loss.

Support Vector Machine Before the rise of deep learning, Support Vector Machines (SVMs) held a prominent position as one of the most favored machine learning models, resulting in a rich of research focused on enhancing the computational efficiency of SVM. For linear SVMs, Joachims (2006) introduces a first-order algorithm that efficiently resolves its Quadratic Programming (QP) problem with nearly linear time complexity. In the case of SVM classification, established algorithms such as SVM-Light Platt (1998a), SMO Platt (1998b), LIBSVM Chang & Lin (2011), and SVM-Torch Collobert & Bengio (2001) excel in high-dimensional data settings.

It is well-known that attention maps, represented as softmax outputs, serve as a mechanism for selecting relevant features and reveal the tokens relevant to classification. Tarzanagh et al. (2023b;a) establish the connection between attention and SVM. Tarzanagh et al. (2023a) formulate the attention computation as $X_i^\top V \mathbb{S}(X_i K Q^\top z_i)$, where $\mathbb{S}$ is the softmax function, X_i is the i-th input sentence and z_i is the classification token $[\text{CLS}]$. They demonstrate that one-layer transformer solves an SVM problem that separates the optimal tokens within each input sequence from other tokens. However, their approach relies on assumptions concerning the loss function and token sequences. In our paper, we relax their assumption of the loss function's derivative and Lipschitz. In addition, they provide proofs for the regularization path analysis in casual language modeling task. In our paper, we show that the $QK^\top$ trained by gradient descent also converge to the SVM solution with Frobenius norm object in casual language modeling task.

In Nguyen et al. (2022), they establish a connection between self-attention and support vector regression (SVR) by deriving self-attention as a support vector expansion. They introduce a principled primal-dual framework for the study and development of self-attentions. Through the solution of a support vector regression problem, they achieved a more profound comprehension and elucidation of diverse attention mechanisms.

3 The Equivalence Between Softmax attention and Support Vector Machine

In this section, we first provide several important definitions and then our main theorem.

3.1 Preliminary

Notations Let $u \in \mathbb{R}^n, \exp(u) \in \mathbb{R}^n$ denote the vector that $\exp(x)_i = \exp(x_i)$. Given positive integer n, we use $[n]$ to denote set $\{1, 2, \cdots, n\}$. For two vectors u, v, we use $\langle u, v \rangle$ to denote the inner product. Let $\mathbf{1}_n$ denote a length-n vector where all the entries are ones. For matrix $A \in \mathbb{R}^{n \times d}$, we use $A_{*,i}$ to denote the i-the column of matrix A for each $i \in [d]$. We use $u \circ v$ to denote a vector whose i-th entry is $u_i v_i$.

Let's define a vector x in $\mathbb{R}^{n^2}$ and a matrix X in $\mathbb{R}^{n \times n}$. We say that x is the vectorization of X, denoted as $x = \mathrm{vec}(X)$, if the i-th row of matrix X is equivalent to the subsequence of elements in x from the $(i-1)n+1$-th position to the in-th position, for all i in the range $[n]$. Conversely, if we have a vector x and we want to reconstruct the matrix $X, X = \mathrm{mat}(x)$. Additionally, for two matrices A in $\mathbb{R}^{n_1 \times d_1}$ and B in $\mathbb{R}^{n_2 \times d_2}$, the Kronecker product $A \otimes B$ results in a new matrix in $\mathbb{R}^{n_1 n_2 \times d_1 d_2}$. Each entry at position $(i_1 - 1)n_2 + i_2, (j_1 - 1)d_2 + j_2$ in this new matrix is obtained by multiplying the corresponding elements from A and B, where $i_1 \in [n_1], j_1 \in [d_1], i_2 \in [n_2], j_2 \in [d_2]$.

In this paper, we denote m as the number of data points. Let n denote the length of sentence. Let d denote the size of feature dimension.

We first give the formal definitions of some basic functions in attention computation. We first introduce the computation of softmax with key and value matrix. By using the tensor trick, we turn the multiple regression into a single regression with re-ordering all the entries.

Definition 3.1. *Let $A_{l_0,1}, A_{l_0,2} \in \mathbb{R}^{n \times d}$. Let $\mathsf{A}_{l_0} = A_{l_0,1} \otimes A_{l_0,2} \in \mathbb{R}^{n^2 \times d^2}$. Let $\mathsf{A}_{l_0,j_0} \in \mathbb{R}^{n \times d^2}$ denote the j_0-th block of $\mathsf{A}_{l_0} \in \mathbb{R}^{n^2 \times d^2}$. Let $x = \mathrm{vec}(X) \in \mathbb{R}^{d^2}$*

For each $l_0 \in [m]$, for each $j_0 \in [n]$.

We define $u(x)_{l_0,j_0} \in \mathbb{R}^n$ as follows

$$\underbrace{u(x)_{l_0,j_0}}_{n \times 1} := \exp(\underbrace{\mathsf{A}_{l_0,j_0}}_{n \times d^2} \underbrace{x}_{d^2 \times 1})$$

Definition 3.2. *For each $l_0 \in [m]$, for each $j_0 \in [n]$.*

We define $\alpha(x)_{l_0,j_0} \in \mathbb{R}$ as follows

$$\underbrace{\alpha(x)_{l_0,j_0}}_{\text{scalar}} := \langle \underbrace{u(x)_{l_0,j_0}}_{n \times 1}, \underbrace{\mathbf{1}_n}_{n \times 1} \rangle.$$

Next, we give the formal definition of

Definition 3.3. *Let $A_{l_0,1}, A_{l_0,2} \in \mathbb{R}^{n \times d}$. Let $\mathsf{A}_{l_0} = A_{l_0,1} \otimes A_{l_0,2} \in \mathbb{R}^{n^2 \times d^2}$. Let $\mathsf{A}_{l_0,j_0} \in \mathbb{R}^{n \times d^2}$ denote the j_0-th block of $\mathsf{A}_{l_0} \in \mathbb{R}^{n^2 \times d^2}$.*

For each $l_0 \in [m]$, for each $j_0 \in [n]$, we define $f(x)_{l_0,j_0} : \mathbb{R}^{d^2} \to \mathbb{R}^n$,

$$\underbrace{f(x)_{l_0,j_0}}_{n \times 1} := \underbrace{\alpha(x)_{l_0,j_0}^{-1}}_{\text{scalar}} \cdot \underbrace{u(x)_{l_0,j_0}}_{n \times 1}$$

Definition 3.4. *For each $l_0 \in [m]$, for each $i_0 \in [d]$, we define $h(y)_{l_0,i_0} \in \mathbb{R}^{d^2} \to \mathbb{R}^n$*

$$\underbrace{h(y)_{l_0,i_0}}_{n \times 1} = \underbrace{A_{l_0,3}}_{n \times d} \underbrace{y_{i_0}}_{d \times 1}$$

Here $y_{i_0} \in \mathbb{R}^d$ is i_0-th column of $y \in \mathbb{R}^{d \times d}$. (We can view y as the value matrix V)

By using the above definitions, we formulate the attention as $\langle f(x)_{l_0,j_0}, h(y)_{l_0,i_0}\rangle$. Next, we formally define the logistic loss function.

Definition 3.5. *Let $x \in \mathbb{R}$, then we defined the logistic function as follows:*

$$g(x) := \frac{1}{1+\exp(-x)}$$

Now, we provide the formal definition of empirical loss function.

Definition 3.6. *If the following conditions hold*

- *Let $f(x)_{l_0,j_0}$ be defined in Definition 3.3*
- *Let $h(y)_{l_0,i_0}$ be defined in Definition 3.4*
- *Let $\theta \in \mathbb{R}$*
- *Let $g : \mathbb{R} \to \mathbb{R}$ denote logistic function which follows from Definition 3.5*
- *Let $b_{l_0,j_0,i_0} \in \mathbb{R}$*

Then we define the loss function based on logistic function as follows:

$$L(x,y)_{l_0,j_0,i_0} := g(\langle f(x)_{l_0,j_0}, h(y)_{l_0,i_0}\rangle b_{l_0,j_0,i_0})$$

Then,

Definition 3.7 (Algorithm W-GD). *Given $X(0) \in \mathbb{R}^{d\times d}$, for $k \geq 0$ do:*

$$X(k+1) = X(k) - \eta \nabla L(X(k))$$

Now, we introduce a convex hard-margin SVM problem, denoted as (Att-SVM). Its objective is to distinguish a particular token from the other tokens in the input sequence A_{l_0} by evaluating the dot product between the key-query features before applying the softmax.

Definition 3.8. *Given $A_{l_0} \in \mathbb{R}^{n\times d}$, we defined the Att-SVM as the following:*

$$\begin{aligned} X^{\mathrm{mm}} = &\arg\min_X \|X\|_F \\ &\textit{s.t. } (A_{l_0,\mathrm{opt}_i} - A_{l_0,t})^\top X A_{l_0,i} \geq 1 \quad \textit{for all } \ t \neq \mathrm{opt}_i, i \in [n], l_0 \in [m] \end{aligned}$$

3.2 Asymptotically Equivalent Convergence Direction of Softmax Attention

Before we introduce our main result, we give the basic definitions of optimal tokens and support indices in our casual language modeling setting.

First, we define the score of the tokens and the optimal tokens. Tokens' scores can provide valuable information of the importance of individual tokens and how they contribute to the overall objective respectively. The tokens' scores quantifies the impact of each token to specific classification or prediction task. The optimal token is the token that manifest the greatest relevance to the input sequence.

Definition 3.9 (Token Score and Optimality). *Given a prediction head $v_i \in \mathbb{R}^d$, the score of a token $A_{l_0 t}$ of input A_{l_0} is defined as $\gamma_{l_0 t} = B_{l_0,} \cdot v_i^\top A_{it}$. The optimal token for each input A_{l_0} is given by the index* $\mathrm{opt}_{l_0} \in \arg\max_{t\in[T]} \gamma_{l_0 t}$ *for all $l_0 \in [m]$.*

Next, we state two assumptions that is of significant importance to guarantee that the attention layer possess a benign optimization landscape. Specifically, the first part of the assumption below provide insights into overparameterization. The second assumption described a scenario that every token that is not optimal has the same token score, which is less than the score of the optimal token. In the scenarios where data are distributed such that d is sufficiently large, such phenomenon is likely to persist. The second assumption states that for any token that is not optimal, possess the same token score. The second part of the assumption below is a relatively stringent assumption that needs to be relaxed.

Assumption 3.10 (Assumption B in Tarzanagh et al. (2023a)). *Optimal tokens' indices $(\mathtt{opt}_{l_0})_{l_0=1}^m$ are unique and one of the following on the tokens holds:*

1. *All tokens are support vectors, i.e., $(x_{i\mathtt{opt}_i} - x_{it})^\top W^{mm} z_i = 1$ for $\forall t \neq \mathtt{opt}_i$ and $i \in [n]$.*
2. *The token's scores, as defined in Definition 3.9, satisfy $\gamma_{it} = \gamma_{i\tau} < \gamma_{i\mathtt{opt}_i}$ for $\forall t, \tau \neq \mathtt{opt}_i$ and $i \in [n]$.*

The next lemma is a important intermediate step of analyzing the loss function defined in Definition 3.6. It computes the gradient of the loss function whose lipschitz property is of great importance of proving our main results and would be evaluated in the lemmas afterwards.

Lemma 3.11 (Formal version of Lemma F.2). *If the following conditions hold*

- *Let $L(x, y)_{l_0,j_0,i_0}$ be defined as Definition 3.6*
- *Let $f(x)_{l_0,j_0}$ be defined in Definition 3.3*
- *Let $h(y)_{l_0,i_0}$ be defined in Definition 3.4*

Then we have

$$\begin{aligned}
&\frac{\mathrm{d}L(x, y)_{l_0,j_0,i_0}}{\mathrm{d}x_i} \\
&= g(\langle f(x)_{l_0,j_0}, h(y)_{l_0,i_0}\rangle)(1 - g(\langle f(x)_{l_0,j_0}, h(y)_{l_0,i_0}\rangle))b_{l_0,j_0,i_0} \\
&\quad \cdot (\langle f(x)_{l_0,j_0} \circ \mathsf{A}_{l_0,j_0,i}, h(y)_{l_0,i_0}\rangle - \langle f(x)_{l_0,j_0}, h(y)_{l_0,i_0}\rangle\langle f(x)_{l_0,j_0}, \mathsf{A}_{l_0,j_0,i}\rangle)
\end{aligned}$$

Next, we would prove a well known fact in the real world applications regarding the logistic function.

Lemma 3.12 (Informal version of Lemma F.4). *Let $g(x)$ be defined in Definition 3.5. Then we have*

$$|g'(x) - g'(\widehat{x})| \leq |x - \widehat{x}|$$

Then, we evaluate the lipschitz property of several basic function, which is the stepping stone for many analysis in this paper. By combining those analysis and Lemma 3.12, we are able to evaluate the lipschitz property of $\nabla L(x, :)$, stated as the lemma below.

Lemma 3.13 (Informal version of F.9). *If the following conditions hold*

- *Let $f(x)_{l_0,j_0}$ be defined in Definition 3.3*
- *Let $h(y)_{l_0,i_0}$ be defined in Definition 3.4*
- *Let $d(x) := \langle f(x)_{l_0,j_0}, h(y)_{l_0,i_0}\rangle$*
- *Let $R \geq 4$*
- *Let $x, y \in \mathbb{R}^d$ satisfy $\| \mathsf{A}_{l_0,j_0} x\|_2 \leq R$ and $\| \mathsf{A}_{l_0,j_0} y\|_2 \leq R$*
- *$\| \mathsf{A}_{l_0,j_0} \| \leq R$*
- *Let $L(x, y)_{l_0,j_0,i_0}$ be defined in Definition 3.5*
- *Let $w(x) := \langle f(x)_{l_0,j_0}, v_1\rangle - \langle f(x)_{l_0,j_0}, v_2\rangle\langle f(x)_{l_0,j_0}, v_3\rangle$*

Then we have

$$|\nabla L(x, :)_{l_0,j_0,i_0} - \nabla L(\widehat{x}, :)_{l_0,j_0,i_0}| \leq 3n^3 R^7 \exp(13R^2)\|x - \widehat{x}\|_2$$

This lemma is the basis to proving our main results.

Finally, we state the main result of this paper. The theorem below proved that the global convergence of the gradient descent algorithm to the max-margin direction X^{mm} under the second assumption of Assumption 3.10 which states that all the tokens who are not optimal possess the same score that is lower than the score of the optimal token.

Theorem 3.14 (Informal version of Theorem G.4)**.** *Suppose Assumption 3.10 on the tokens' score hold. Let $X(k)$ denote the k-th iteration of X. Then, Algorithm W-GD (Definition 3.7) with the step size $\eta \leq 1/L_X$ and any starting point $X(0)$ satisfies*

$$\lim_{k \to \infty} \frac{X(k)}{\|X(k)\|_F} = \frac{X^{mm}}{\|X^{mm}\|_F}$$

We provide another theorem below. This theorem is a relaxation of the second assumption in Assumption 3.10. This theorem demonstrates that the global convergence can still happen even when the score of the tokens are equal.

Theorem 3.15 (Informal version of Theorem G.5)**.** *For any initialization $X(0)$, there exists a dataset dependent sufficiently small $\delta > 0$ such that the following holds: Suppose non-optimal scores obey $|\gamma_{it} - \gamma_{i\tau}| \leq \delta$ for all $t, \tau \neq opt_i, i \in [m]$. Then, Algorithm X-GD, with $\eta \leq 1/(2L_x)$ obeys*

$$\lim_{k \to \infty} \|X(k)\|_F = \infty \text{ and } \lim_{k \to \infty} \frac{X(k)}{\|X(k)\|_F} = \frac{X^{mm}}{\|X^{mm}\|_F}$$

By showing that gradient descent converges to the optimal solution for SVM with Frobenius norm objective, we establish a fundamental connection between attention mechanisms and Att-SVM, shedding light on the optimization dynamics of attention layers. Specifically, we demonstrate that the weight matrix, denoted as W, learned through gradient descent during language modeling pretraining, converges to a solution analogous to that of SVM with a Frobenius norm objective. Notably, our investigations reveal that during this optimization process, only a few selected tokens can be equated to feature vectors and play a pivotal role in the gradient computation for each update, while tokens that do not function as support vectors have gradients that effectively amount to zero. This intrinsic sparsity in attention mechanisms paves the way for the practical applicability of sparse training techniques.

Token Selection & Separation In Zhang et al. (2023), they introduce a groundbreaking approach aimed at optimizing the memory utilization of the KV cache. This innovative method yields substantial reductions in memory footprint, a development that can have far-reaching implications for various applications. Their approach lies in the observation that only a small subset of tokens significantly contributes to the value computation during the process of attention scoring.

Considering the language modeling task we study in this paper, our result provide the evidence to this phenomenon. Attention mechanisms are tasked with the responsibility of identifying a highly relevant subset of tokens from the input sequence. This selection process is integral to the accurate prediction of subsequent tokens. Consequently, for tokens that exhibit low token scores (as defined in Definition 3.9), omitting their computational contributions does not detrimentally impact the final outcome. This unique insight underscores the intriguing interplay between attention mechanisms and the principles underlying Support Vector Machines (SVM). This insight underscores the interplay between attention mechanisms and SVM.

3.3 Lipschitz of Basic Functions

For the ease of proving the lipschitz property for complex terms, we would like to address the lipschitz property of some basic terms as follows:

- $\| \exp(\mathsf{A}_{l_0,j_0} x) - \exp(\mathsf{A}_{l_0,j_0} y)\|_2 \leq R \exp(R^2) \cdot \|x - y\|_2$
- $|\alpha(x)_{l_0,j_0} - \alpha(y)_{l_0,j_0}| \leq \sqrt{n} \cdot \| \exp(Ax) - \exp(Ay)\|_2$
- $|\alpha(x)_{l_0,j_0}^{-1} - \alpha(y)_{l_0,j_0}^{-1}| \leq \beta^{-2} \cdot |\alpha(x) - \alpha(y)|$
- $\|f(x)_{l_0,j_0} - f(y)_{l_0,j_0}\|_2 \leq R_f \cdot \|x - y\|_2$
- $\|c(x,z)_{l_0,j_0,i_0} - c(y,z)_{l_0,j_0,i_0}\|_2 \leq R^2 \beta^{-2} n \exp(3R^2) \|x - y\|_2$
- $\| \operatorname{diag}(f(x)_{l_0,j_0}) - \operatorname{diag}(f(y)_{l_0,j_0})\| \leq \beta^{-2} n \exp(3R^2) \|x - y\|_2$
- $f(x)_{l_0,j_0} f(x)_{l_0,j_0}^\top - f(y)_{l_0,j_0} f(y)_{l_0,j_0} \leq 2\beta^{-3} n^2 \exp(5R^2) \|x - y\|_2$

The lipschitz property for these basic function is easy to prove, and we will use them as the stepping stone for proving the lipschitz property for $\nabla L(x, :)$.

3.4 Lipschitz for Logistic Loss

By using the lipschitz property of several basic functions, we could combine them to prove the lipschitz property for more complex functions. In this section, we aim to find M such that

$$|\nabla L(x,:)_{l_0,j_0,i_0} - \nabla L(\widehat{x},:)_{l_0,j_0,i_0}| \le M \cdot \|x-\widehat{x}\|_2$$

where

$$L(x,:)_{l_0,j_0,i_0} = \mathrm{logistic}(\langle f(x)_{l_0,j_0}, h(y)_{l_0,i_0}\rangle b_{l_0,j_0,i_0})$$

denote the loss function based on $\mathrm{logistic}(.)$. We find M by the following steps:

Step 1: Prove the lipschitz property for logistic function we can prove the following property for $\mathrm{logistic}(.)$ through mean value theorem:

$$|\,\mathrm{logistic}(x) - \mathrm{logistic}(\widehat{x})| \le |x-\widehat{x}|$$

Step 2: Compute the gradient $\nabla L(x,:)_{l_0,j_0,i_0}$. We are able to compute

$$\begin{aligned}&\frac{\mathrm{d}L(x,y)_{l_0,j_0,i_0}}{\mathrm{d}x_i}\\ =\,& g(\langle f(x)_{l_0,j_0}, h(y)_{l_0,i_0}\rangle)(1-g(\langle f(x)_{l_0,j_0}, h(y)_{l_0,i_0}\rangle))b_{l_0,j_0,i_0}\\ &\cdot(\langle f(x)_{l_0,j_0}\circ \mathsf{A}_{l_0,j_0,i}, h(y)_{l_0,i_0}\rangle - \langle f(x)_{l_0,j_0}, h(y)_{l_0,i_0}\rangle\langle f(x)_{l_0,j_0}, \mathsf{A}_{l_0,j_0,i}\rangle)\end{aligned}$$

Step 3: Reform $\nabla L(x,:)_{l_0,j_0,i_0}$ for the convenience of analysis We reform the gradient to

$$\frac{\mathrm{d}L(x,y)_{l_0,j_0,i_0}}{\mathrm{d}x_i} = g'(\langle f(x)_{l_0,j_0}, h(y)_{l_0,i_0}\rangle)b_{l_0,j_0,i_0}\cdot(\langle f(x)_{l_0,j_0}, v_1\rangle - \langle f(x)_{l_0,j_0}, v_2\rangle\langle f(x)_{l_0,j_0}, v_3\rangle)$$

where

- $v_1 := h(y)_{l_0,i_0}\circ \mathsf{A}_{l_0,j_0,i}$
- $v_2 := h(y)_{l_0,i_0}$
- $v_3 := \mathsf{A}_{l_0,j_0,i}$

this would make the analysis for the gradient more convenient.

Step 4: Split the gradient and combine the lipschitz for basic functions to get the final result

$$M = 3n^3R^7\exp(13R^2)$$

4 Experiment

To validate the Theorem 3.15, we use a 1-layer casual transformer to perform the casual language modeling task with a synthetic dataset, and compare the cosine similarity of $X(k)$ (Definition 3.6) and X^{mm} (Definition 3.8) during the optimization of $X(k)$ with logistic loss. We first design a synthetic language and specify all token-level autoregressive sample probability, and we then sample m sequences with sequence length n from this synthetic language. The Att-SVM solution X^{mm} is solved by treating each sub-sequence next-token prediction problem $P(A_{l_0,j_0}|[A_{l_0,1}A_{l_0,2}\dots A_{l_0,j_0-1}])$ as an individual classification problem over the vocabulary set with query from the last token A_{l_0,j_0-1}. We also add a soft margin term to Attn-SVM minimization problem since the vanilla formulation of X^{mm} is often unsolvable due to the non-existence of linear separability of tokens.

Definition 4.1. *Given $A_{l_0,1}\in\mathbb{R}^{n\times d}$, we defined the soft-margin Att-SVM as the following:*

$$\begin{aligned}X^{\mathrm{mm}}_{\textit{soft}} =\ & \arg\min_X \|X\|_F + C\sum_{l_0,t}\xi_{l_0,t}\\ & \textit{s.t.}\ (A_{l_0,\mathrm{opt}_i} - A_{l_0,t})^\top X A_{l_0,i} \ge 1-\xi_{l_0,t}\ \ \&\ \ \xi_{l_0,t}\ge 0\quad \textit{for all}\ \ t\ne \mathrm{opt}_i, i\in[n], l_0\in[m]\end{aligned}$$

Although we relax the hard-margin Att-SVM (Definition 3.8) to a soft-margin one, we empirically find using a large penalty term $C = 10$ still works well. This observation still complies with our Theorem 3.15 and is further illustrated in Figure 2 which shows that the softmax attention is implicitly maximizing the margin between optimal and non-optimal tokens throughout the training process.

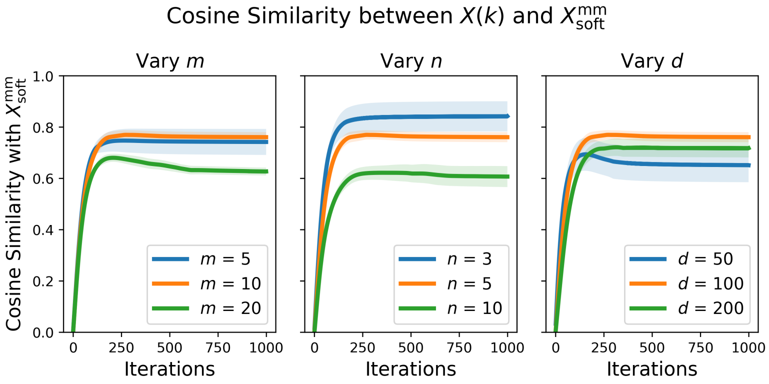

Figure 1: Synthetic Dataset: Casual Language Modeling with Default Configuration as $m = 10, n = 5, d = 100$. We control over m, n, and d in each subfigure and report the mean, the colored line, and standard error, the shaded areas, of the cosine similarity during the optimization of $X(k)$ with a known $X_{\text{soft}}^{\text{mm}}$ solution over 5 different runs.

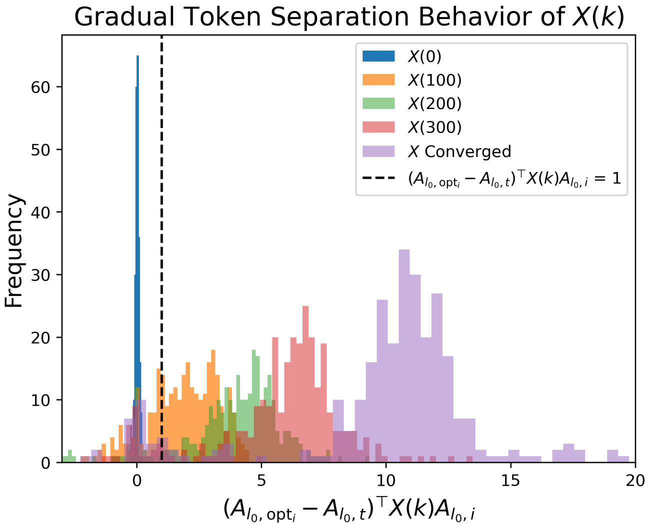

Figure 2: Histogram of Token Separation Margin between Optimal Tokens and Non-optimal Tokens of Softmax Attention in the Synthetic Casual Language Modeling Task. k here is the number of gradient descent steps. The initial margin is near 0 as shown by the blue bins, and as training proceeds, the margin generally becomes larger and eventually converges to violet bins as sufficiently large positive values.

5 Conclusion

In this work, we build a profound connection between attention mechanisms and Att-SVM for casual language modeling task. We've shown that the weight matrix $W = QK^\top$, trained via gradient descent in language modeling pretraining, converges to the SVM solution with a Frobenius norm objective. Furthermore, our experiments have revealed the practical implications of this connection, including token selection and contextual sparsity, where only select tokens contribute to gradients, token separation during training, and the stability of training with softmax attention compared to linear attention. These findings not only deepen our understanding of attention mechanisms but also open new avenues for enhancing training efficiency and exploring novel applications in natural language processing.

REFERENCES

Josh Alman and Zhao Song. Fast attention requires bounded entries. In *NeurIPS*, 2023.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, March 2022. doi: 10.1162/coli_a_00422. URL https://aclanthology.org/2022.cl-1.7.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. On the Ability and Limitations of Transformers to Recognize Formal Languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7096–7116, Online, November 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.576. URL https://aclanthology.org/2020.emnlp-main.576.

Satwik Bhattamishra, Arkil Patel, and Navin Goyal. On the computational power of transformers and its implications in sequence modeling. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pp. 455–475, Online, November 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.conll-1.37. URL https://aclanthology.org/2020.conll-1.37.

Jan van den Brand, Zhao Song, and Tianyi Zhou. Algorithm and hardness for dynamic attention maintenance in large language models. *arXiv preprint arXiv:2304.02207*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

Beidi Chen, Tri Dao, Eric Winsor, Zhao Song, Atri Rudra, and Christopher Ré. Scatterbrain: Unifying sparse and low-rank attention. *Advances in Neural Information Processing Systems (NeurIPS)*, 34: 17413–17426, 2021a.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021b.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Ronan Collobert and Samy Bengio. Svmtorch: Support vector machines for large-scale regression problems. *Journal of machine learning research*, 1(Feb):143–160, 2001.

Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10578–10587, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Huaian Diao, Zhao Song, Wen Sun, and David Woodruff. Sketching for kronecker product regression and p-splines. In *International Conference on Artificial Intelligence and Statistics*, pp. 1299–1308. PMLR, 2018.

Huaian Diao, Rajesh Jayaram, Zhao Song, Wen Sun, and David Woodruff. Optimal sketching for kronecker product regression and low rank approximation. *Advances in neural information processing systems*, 32, 2019.

Javid Ebrahimi, Dhruv Gelda, and Wei Zhang. How can self-attention networks recognize Dyck-n languages? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4301–4306, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.384. URL https://aclanthology.org/2020.findings-emnlp.384.

Yeqi Gao, Zhao Song, Weixin Wang, and Junze Yin. A fast optimization view: Reformulating single layer attention in llm based on tensor and svm trick, and solving it in matrix multiplication time. *http://arxiv.org/abs/2309.07418*, 2023a.

Yeqi Gao, Zhao Song, and Shenghao Xie. In-context learning for attention scheme: from single softmax regression to multiple softmax regression via a tensor trick. *arXiv preprint arXiv:2307.02419*, 2023b.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275.

Thorsten Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 217–226, 2006.

Shuai Li, Zhao Song, Yu Xia, Tong Yu, and Tianyi Zhou. The closeness of in-context learning and weight shifting for softmax regression. *arXiv preprint arXiv:2304.13276*, 2023.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, and Beidi Chen. Deja vu: Contextual sparsity for efficient llms at inference time. In *Manuscript*, 2023.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. *arXiv preprint arXiv:2305.17333*, 2023.

Tan Minh Nguyen, Tam Minh Nguyen, Nhat Ho, Andrea L Bertozzi, Richard Baraniuk, and Stanley Osher. A primal-dual framework for transformers and neural networks. In *The Eleventh International Conference on Learning Representations*, 2022.

OpenAI. Openai: Introducing chatgpt, 2022.

OpenAI. Gpt-4 technical report, 2023.

Abhishek Panigrahi, Sadhika Malladi, Mengzhou Xia, and Sanjeev Arora. Trainable transformer in transformer. *arXiv preprint arXiv:2307.01189*, 2023.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International conference on machine learning*, pp. 4055–4064. PMLR, 2018.

J Platt. Making large-scale support vector machine learning practical, 1998a.

John Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998b.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. ., 2018.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68, 2021.

Davoud Ataee Tarzanagh, Yingcong Li, Christos Thrampoulidis, and Samet Oymak. Transformers as support vector machines. *arXiv e-prints*, pp. arXiv–2308, 2023a.

Davoud Ataee Tarzanagh, Yingcong Li, Xuechen Zhang, and Samet Oymak. Margin maximization in attention mechanism. *arXiv preprint arXiv:2306.13596*, 2023b.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*, 2019.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Haixu Wu, Jialong Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Flowformer: Linearizing transformers with conservation flows. *arXiv preprint arXiv:2202.06258*, 2022.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

Shunyu Yao, Binghui Peng, Christos Papadimitriou, and Karthik Narasimhan. Self-attention networks can process bounded hierarchical languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3770–3785, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.292. URL https://aclanthology.org/2021.acl-long.292.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ByxRM0Ntvr.

Amir Zandieh, Insu Han, Majid Daliri, and Amin Karbasi. Kdeformer: Accelerating transformers via kernel density estimation. *arXiv preprint arXiv:2302.02451*, 2023.

Lichen Zhang. Speeding up optimizations via data structures: Faster search, sample and maintenance. Master's thesis, Carnegie Mellon University, 2022.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022a.

Yi Zhang, Arturs Backurs, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, and Tal Wagner. Unveiling transformers with lego: a synthetic reasoning task, 2022b. URL https://arxiv.org/abs/2206.04301.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H _2 o: Heavy-hitter oracle for efficient generative inference of large language models. *arXiv preprint arXiv:2306.14048*, 2023.

Haoyu Zhao, Abhishek Panigrahi, Rong Ge, and Sanjeev Arora. Do transformers parse while predicting the masked word? *arXiv preprint arXiv:2303.08117*, 2023.